Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1-18. (Canceled)

19. (Currently amended) A method executed in a processor comprising:

loading a ~~first number N~~ plurality of groups of bits from a ~~second extended multimedia register~~ source into ~~a lower half of a 2N wide bit first extended multimedia~~ a plurality of non-contiguous groups of bits of a destination register; and

duplicating ~~in a upper half of the 2N bit wide first extended multimedia~~ the plurality of non-contiguous groups of bits in the destination register into subsequent groups of bits in the destination register.

20. (Currently amended) The method of claim 19 in which the source is an ~~second~~ extended multimedia register ~~is a memory location and where N is 64 bits~~.

21. (Currently amended) The method of claim 20 in which the extended multimedia register is ~~memory location~~ configured as ~~contains~~ a double floating point data type.

22. (Currently amended) The method of claim 19 in which the ~~second extended multimedia register~~ source is a memory location ~~128 bit source register and N is 64 bits~~.

23. (Currently amended) The method of claim ~~19~~ 22 in which the ~~128 bit source register contains~~ memory location is configured as a double floating point data type.

24-72. (Canceled)

73. (Currently amended) A processor comprising:

basic program registers;

an address space;

floating point unit (FPU) registers;

~~single instruction multiple data (SIMD) extension registers;~~

a ~~first extended multimedia~~ source ~~register~~;

a ~~second extended multimedia~~ destination register; and

logic to load a ~~first portion~~ plurality of groups of bits of the ~~second extended multimedia~~ source ~~register~~ into a ~~first portion~~ plurality of non-contiguous groups of bits in the ~~of the first extended multimedia~~ destination register and duplicate the plurality of non-contiguous groups ~~that first portion~~ of bits ~~in a~~ into subsequent ~~portion~~ groups of bits in the ~~of the first extended multimedia~~ destination register.

74. (Currently Amended) The processor of claim 73 in which the ~~first portion of the second extended multimedia~~ source ~~register~~ is an extended multimedia register ~~64 bits~~ representing a double floating point data type ~~in a memory location~~.

75. (Currently Amended) The processor of claim 73 in which the ~~first portion of the second extended multimedia~~ source register ~~is 64 bits representing~~ represents a double floating point data type ~~in another source register~~ in a memory location.

76. (Currently Amended) The processor of claim 73 in which the ~~first portion of the first extended multimedia~~ destination register is an extended multimedia register ~~loaded with bits [63-0] of the first portion of the second extended multimedia register and the subsequent portion of the first extended multimedia register is loaded with bits [63-0] of the first portion of the second extended multimedia register~~.

77-82. (Cancelled)

83. (New) A computer instruction comprising:

a load and duplicate instruction that causes a processor to load a plurality of groups of bits from a source into a plurality of non-contiguous groups of bits of a destination register and duplicate the plurality of non-contiguous groups of bits in the destination register into subsequent groups of bits in the destination register.

84. (New) The instruction of claim 83 in which the source is an extended multimedia register.

85. (New) The instruction of claim 84 in which the extended multimedia register is configured as a double floating point data type.

86. (New) The instruction of claim 83 in which the source is a memory location.

87. (New) The instruction of claim 86 in which the memory location is configured as a double floating point data type.

88. (New) A computer program product residing on a computer readable medium having instructions stored thereon which, when executed by a processor, cause the processor to:

load a plurality of groups of bits from a source into a plurality of non-contiguous groups of bits of a destination register; and

duplicate the plurality of non-contiguous groups of bits in the destination register into subsequent groups of bits in the destination register.

89. (New) The computer program product of claim 88 in which the source is an extended multimedia register.

90. (New) The computer program product of claim 89 in which the extended multimedia register is configured as a double floating point data type.

91. (New) The computer program product of claim 88 in which the source is a memory location.

92. (New) The computer program product of claim 91 in which the memory location is configured as a double floating point data type.